Deception Detection in Videos using Robust Facial Features

Anastasis Stathopoulos[1], Ligong Han[1], Norah Dunbar[2], Judee K. Burgoon[3], and Dimitris Metaxas[1]

[1] Rutgers University,
[2] UC Santa Barbara
[3] University of Arizona

Abstract. In this paper, we approach the problem of deception detection in videos. Current approaches are limited since they (i) are used in short videos focusing only on a small act of deception, (ii) are hard to interpret, and (iii) do not make use of any human model that could help them in the detection task. To address those limitations, we propose a novel framework that uses as input the 1-dimensional Facial Action Unit (FAU) and Gaze signals. By using a higher-level input and not the raw video, we are able to train a conceptually simple, modular and powerful model that achieves state-of-the-art performance in video-based deception detection. Finally, we propose a novel approach to interpret our model's predictions, by computing the attention of the neural network in the time domain. This method can enable domain scientists perform retrospective analysis of deceptive behavior.

Keywords: Video Classification, Explainable AI, Deception Detection

1 Introduction

Whenever people communicate, deception is a reality. It is present in our daily lives and can take many forms. Its detection is of paramount importance not just on a personal level, but for our society as a whole. For instance, accurate deception detection could help law enforcement officials perform their duty more effectively or aid in airport security screening procedures. Therefore, the development of robust automated deception detection (ADD) systems is a long sought-after goal.

Nonetheless, current works at ADD seem limited for the following reasons: (i) they focus on a single act of deception in a really short video, (ii) the interpretability of the used models seems narrow, (iii) they do not utilize any model of the human face or body (such as [1]) to aid the detection task. Since the input to such systems contains humans, modeling of faces or bodies could provide very useful cues, while reducing the risk that the model overfits to background noise and learns unimportant features.

In this paper, our aim is to address these limitations. For this reason, we propose a novel system to detect deception in videos. The deception detection task can actually be modelled as binary video classification. That is, we should predict a positive label for a video that contains a person manifesting deceptive behavior and a negative label when that person is acting truthfully.

However, according to the Interpersonal Deception Theory [2], deception is a dynamic process, in which deceivers adjust their behavior according to how much they think they are being suspected by others. For this reason, we claim that the datasets that contain short video clips focusing on only a single act of deception are not enough for modelling deceptive behavior.

To address this (limitation (i) of current works), we use a dataset that contains large videos (1 hour long) of people playing a version of the board game *The Resistance Game*, a social role-playing game that involves deductive reasoning. Players in the Resistance Game are randomly given one of two roles, deceivers or truth-tellers. The majority of players in a game are truth-tellers. Deceivers know the role of everyone in the game, whereas truth-tellers do not. The game relies on deceivers trying to hide their identity and attempt to prevent the larger group from working together to reveal the deceivers among the group. More information is shown in the Experiments Section.

Research suggests that when people communicate, the non-verbal cues, and especially the face, transfer more social meaning than the verbal ones [3]. Facial expressions can convey a lot of information about one's physical and emotional state. People rely on facial expressions to "collect" both intentional and unintentional meaning during interactions. In an attempt to study facial expressions, the Facial Action Coding System (FACS) [4] was developed. It is a systematic way to code facial motion with respect to non-overlapping facial muscle actions called Facial Action Units (FAUs).

With so much communicating by the face, we opt to incorporate facial cues to create a system that detects deception in videos. Our approach has the following pipeline. We fit a morphable model to a subject's face and with the help of a feature extractor, we get high-level information out of the video modality. In particular, for each frame of our input video we compute the intensities of 17 Facial Action Units (FAUs), which we normalize with the parameters of the morphable model that is fitted to the subject's face. This results in 17 person-agnostic FAU intensities. We also compute 2 gaze angles for each frame of the the input video.

Then, we concatenate those 19 1-dimensional signals (17 FAU and 2 gaze signals) channel-wise and feed this signal as input to our video classification component. This is a Temporal Convolutional Network (or TCN) that takes as input the waveform described before. We choose to use those FAU waveforms as a higher-level representation as opposed to raw pixels for the input video. As deception detection datasets are very small, models that operate directly on raw videos are likely to overfit to background noise. We assume that the chosen representation in more clean and robust than raw videos, which we validate experimentally. Furthermore, the chosen representation is complete, since any facial muscle movement can be decomposed to a combination of FAUs.

Finally, we provide a framework for retrospective analysis of deceptive behavior. More specifically, given the predicted class of each video, we use an *Attention Module* to calculate the time regions that contributed substantially to the prediction of the model. If the video was classified as one containing deceptive behavior those regions could be indicative of when deception happened. Domain experts could then observe the FAU signals and how they are correlated in those time regions to gain some insights about deception indicators.

The effectiveness of our approach is validated by comprehensive evaluation in three datasets. We also provide comparisons with the state-of-the-art as well as an ablation study. Our approach surpasses the state-of-the-art method in deception detection, while it is extremely lightweight and modular.

We summarize the contributions of our work as follows.

- We propose a novel framework that achieves state-of-the-art performance on video-based deception detection as tested on three benchmarks.
- Our framework is modular, lightweight and robust to the identity of a person by nature.
- We provide a framework for retrospective analysis of deceptive behavior.

2 Related Work

Video classification architectures. Motivated by the success of CNNs on image-related tasks and of RNNs on sequence modelling, a natural solution to video classification can be to combine them [5]. Another solution for video classification can be feed-forward models that use 3D Convolutions (C3D) [6, 7] to learn spatiotemporal features. Nonetheless, Simonyan and Zisserman noticed that temporal features are hard to learn only by stacking images, and, therefore proposed to train a two-stream ensemble network [8] that utilizes optical flow information. Finally, in [9] a method with sparse sampling to model long-term temporal dependencies is proposed. However, for the task of video-based deception detection, we can not use any of the previous approaches as an off-the-shelf model. Although, they are very successful in action recognition, they seem to overfit to the identity of the person in the video and fail to extract relevant features for the necessary task.

Deception detection from videos. With the introduction of a dataset that contains video clips from real-life court trials [10, 11], several methods for detecting deceptive behavior in videos have been developed. However, the size of the dataset is very small (104 videos are used in practise). As a result, there are approaches reporting that hand-crafted features perform much better than deep features. For instance, in [12] the authors use IDT (Improved Dense Trajectory) as low-level features to train a micro-expression detector that is used along with the IDT features for deception detection. In [13] the authors use a deep learning model that makes video-level predictions by aggregating the predictions made in short snippets sparsely sampled from the input video. The input of the model consists of a video frame capturing appearance features and 5 optical flows maps that model temporal features. To be able to train this model the authors make use of a meta-learning and an adversarial learning [14, 15] module. We opt not to make use of such methods for training, since we want our model to be interpretable.

Temporal convolutional networks. Recently there has been a trend of using feed-forward architectures for sequence modelling instead of RNNs. Those architectures are called Temporal Convolutional Networks (TCNs). Their main component is 1-dimensional *causal* convolutions, meaning that there is no information leakage from future to past time steps. By using dilated convolutions, TCNs can have exponential receptive fields relative to their depth and thus, they are able to model long-term dependencies. Recent works on speech and language modelling [16–18] replace recurrent architectures and make only use of TCNs. In [19] the authors show that TCNs can outperform baseline recurrent architectures across a variety of sequence modelling tasks. TCNs are also being used by the signal processing community in a variety of task, such as blind source separation [20]. We, as well, choose to use a TCN architecture for our task.

3 Method

We propose a novel framework for non-verbal deception detection in videos that consists of two main modules: (i) a feature extractor, (ii) a video classification model (predictor). An overview of our framework can be shown in Figure 1.

3.1 FAU Waveforms

The face can reveal a plethora of signals related to deception. To create a video-based ADD system, it is important to utilize those signals effectively. Thus, we need a way to extract salient information out of the video modality to use in our system. Instead of providing the raw videos, which may contain a lot of noisy information for our task, as input to our system, we choose to input a higher-level representation to make the learning procedure easier and ensure that the model actually learns features relevant to deceptive behavior.

In particular, let $\mathcal{D} = \{\mathbf{v}^{(i)}, y^{(i)}\}_{i=1}^{M}$ denote our dataset containing M video-label pairs. Each video $\mathbf{v}^{(i)}$ in our dataset is actually a tensor of size $T \times H \times W \times 3$. Instead, of using the video tensor as input to our model, we extract frame-wise facial features using OpenFace [21]. More specifically, for each video $\mathbf{v}^{(i)}$, we compute normalized intensities of N (in our experiments, $N = 17$) FAUs $\{x_j^{(i)}\}_{j=1}^{N} \in [0, T]$. Those N signals are concatenated, resulting in

$$\mathbf{x}^{(i)} = \mathop{\Big\|}_{ch=1}^{N} x_{ch}^{(i)} \tag{1}$$

where $\|$ represents channel-wise concatenation.

The FACS [4] is one of the most comprehensive and objective systems for describing facial expressions. If one thinks of Facial Action Units (FAUs) as a basis, any facial muscle movement can be decomposed to a combination of

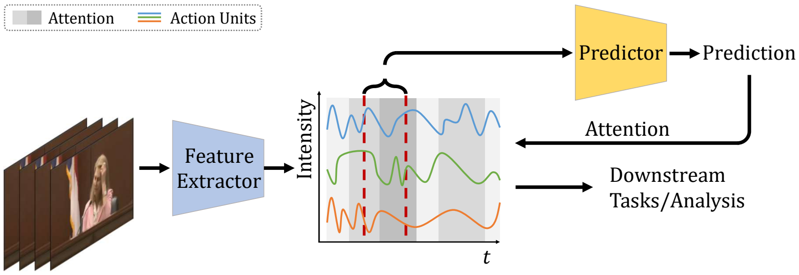

Fig. 1. Illustration of the proposed framework. FAU intensities and Gaze angles are extracted from video sequences. We consider each one of those measurements as an 1-dimensional signal that we normalize. We concatenate those signals channel-wise and use them as input to train a predictor model. Finally, we compute the attention of our model to enable retrospective analysis of deceptive behavior.

FAUs. Therefore, we claim that by replacing $\mathbf{v}^{(i)}$ with $\mathbf{x}^{(i)}$ we keep a less noisy representation of the facial behavior for each subject.

Additional Input. Our framework is quite general, since it can incorporate a variety of different input features, by simply stacking them as extra channels in the input signal. In our model, we choose to use gaze features as well by concatenating the signals of two gaze angles channel-wise with $\mathbf{x}^{(i)}$.

3.2 Video Classification Model

We will now introduce the core module of our method, ie the video classification model. Although, in this model we predict the class of a video, the input to our model is a 1-dimensional signal with 19 channels carrying the appropriate video information and not the raw video itself.

Inspired by the success of Temporal Convolutional Networks in the domains of signal processing [20] and sequence modelling [16–19], we use a TCN for our video classification task. We try to keep the structure of our neural network as simple as possible since we would like our model to be interpretable. Because the input to our model is a waveform that carries high-level features we were able to avoid using deep architectures, in contrast to other deception systems [13] that need to extract the necessary features from the raw video for the classification task.

Base network. The input to our model $\mathbf{x}^{(i)}$ is convolved with 128 1-dimensional kernels of size L to produce 128 1-D features maps. Those features maps are then passed through a ReLU and Batch Normalization [22] layer. The output is then processed by a dense layer implemented as 1-D convolution with kernel size 1.

Residual blocks and classification layer. To be able to capture long-term dependencies in our input we use residual blocks [23] with dilated convolutions [24]. Such a block can be seen in Figure 2. Finally, we average pool the feature maps and apply a Fully Connected layer to get the final prediction.

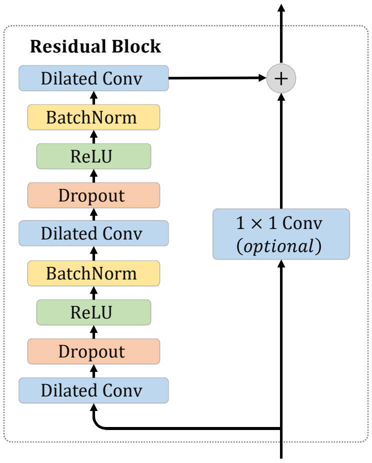

Fig. 2. Residual blocks used in the proposed video classification model.

We want to note that our method can model inputs of arbitrary length and is extremely memory and computation efficient. Furthermore, one can control the extent to how dependencies are captured in the time domain by changing the receptive field of the model. This can be accomplished by stacking more dilated convolutional layers, increasing filter sizes or using larger dilation factors.

More details regarding the implementation of our video-based deception detection model can be found in the next section.

Bayesian ensemble. To boost the performance, we implement a Bayesian neural network (BNNs) variant of the above introduced base model. Briefly, a BNN tries to estimate the posterior over network weights during training. For a classification model, the predictive distribution over labels given input x is $P(y|x) = \mathbb{E}_{P(w|\mathcal{D})}P(y|x,w)$, where $P(y|x,w) = \text{softmax}(f(x|w))$ is the predicted probabilities given a specific feed-forward network f parameterized by weight w. Notice that since estimating the posterior $P(w|\mathcal{D})$ is often intractable [25, 26], variational inference is commonly adopted. The posterior is approximated by $P(w|\mathcal{D}) \approx Q_\theta(w)$ and trained by maximizing the evidence lower bound (ELBO) [27, 28]. Finally, at testing time, the BNN prediction is approximated by Monte Carlo sampling (with K samples),

$$P(y|x;\theta) = \frac{1}{K}\sum_{k=1}^{K} \text{softmax}(\hat{f}^{(k)}(x)). \tag{2}$$

3.3 Attention Module

Given a trained model, it is beneficial and informative to visualize its attention. As opposed to self-attention [29], we use Grad-CAM [30]-like method, which is flexible and does not require architectural modification to the model. As a recap, the Grad-CAM assigns importance to each pixel as the gradient of network output w.r.t. a certain feature layer. For 1-D time series indexed by subscript t, denoting the class score as Y^c and feature map in the k-th channel as F^k, the importance weight can be computed as $\frac{\partial Y^c}{\partial F^k_t}$. Aggregating the importance weights for all pixels, we get the neuron importance weight of the k-th channel for class c, $\alpha^c_k = \frac{1}{Z}\sum_t \frac{\partial Y^c}{\partial F^k_t}$ where Z is the normalization constant (global average pooling). Then the attention map can be obtained from $A^c_{Grad-CAM} = ReLU\left(\sum_k \alpha^c_k F^k\right)$. Motivated by [31], a positive gradient at a specific location implies increasing the pixel intensity in F^k results in a positive impact on the prediction score. As such, we adopted a new channel-weighted attention mechanism A_{ch},

$$A^c_{ch} = \frac{1}{Z} ReLU\left(\sum_k \sum_t ReLU\left(\frac{\partial Y^c}{\partial F^k_t}\right) F^k\right). \tag{3}$$

4 Experiments on Video-Based Deception Detection

In this section, we first introduce more details about our implementation. Then, we present the evaluation setting of our approach, which includes experiments in 3 datasets, and comparisons with the state-of-the-art methods. The datasets used are the *Real-Life Trial* dataset [10] and *Bag-of-Lies* [32], which are available to the public. We also use a dataset that contains very long videos from people playing a version of the board game *The Resistance.* From now on, we will refer to that dataset simply as *The Resistance* dataset. Then, we perform an ablation study on the input components of our method. Finally, with the help of our attention module, we find the *key frames* for every video, ie the frames that contributed the most to the decision of our model. By inspecting the value of our input in those frames, we get valuable insight at what our model "thinks" that constitutes deception as a function of facial movements (FAUs intensities).

4.1 Implementation Details

Since the videos in the *Real-Life Trial* dataset are captured with different frame rates, the time scale of our extracted features is not similar for every video. To handle this inconsistency, we interpolate values in the videos captured with frame rates less than 30 fps. As a result, the waveforms extracted to represent each video should contain 30 values for every second in it. We choose that as the scale of our input and let our model learn to use the scale that best describes the data.

Training. We initialize the weights of our model using the initialization described in [33]. For training our model we sample k values of our input signal corresponding to k consecutive frames in the input video. For the *Real-Life Trial* [10] dataset $k = 180$, while for *Bag-of-Lies* $k = 110$. The value of k is chosen to be the minimum number of frames that a video contains in the corresponding dataset. We train our models for 100 epochs in total, starting with a learning rate of 0.001 and use the method described in [34] to dynamically update the learning rate using the gradient with respect to the learning rate of the update rule itself.

In our preliminary experiments, we observe that inserting residual blocks to our model evaluated in *Real-Life Trial* [10] did not increase the performance of our model. Thus, we only use our base network along with the classifier layer for this dataset and omit any residual blocks. Nonetheless, for the other 2 datasets we observe a small performance boost by using residual blocks, and choose to include them for our evaluation.

Inference. During inference, we split the videos into segments each of which contains k frames and perform a forward pass using our model for each one of them, computing softmax scores for them individually. The final prediction of our model in the averaged softmax scores for all segments.

4.2 Experiments on Real-life Trial

Real-Life Trial [10] is a publicly available database for the evaluation of deception detection models. It consists of 121 videos from real-life court room trials. As the utility of some videos is questionable, the recent approaches of [12] and [13] have opted to include only a subset of the dataset for their experiments. In particular, both methods use only 104 out of the 121 available in the dataset, including 54 deceptive and 50 truthful videos.

Evaluation protocol. For the purpose of evaluation, we perform 10-fold cross validation as suggested in [12] and [13]. However, both works claim that the dataset consists of 58 identities and decide to split the dataset into folds based on identities instead of video samples. Nonetheless, the subset used contains only 42 identities and some subjects appear only in 1 video, while others can appear in more than 10% of the the total videos. This can be problematic as the cross validation folds are very unbalanced and in some cases a fold can contain only 4 videos.

This evaluation procedure can result in distorted results based on the videos in each validation fold. For this reason, we use a more objective procedure, ie we split based on video samples making sure that we have enough videos for validating our model in every fold. Unlike FFCSN [13], our method does not model a subject's appearance and thus it is robust in that setting.

To evaluate our method we compute the average classification accuracy (ACC) and the average area under the precision-recall curve (AUC) across the cross validation folds, as suggested in prior works. Earlier works [10, 11, 35, 36] use the former metric, while most recent ones [12, 13] use the latter to account for the imbalance of the positive and negative classes.

Baselines. We establish 2 baselines to compare with our results. First, we implement a baseline proposed in Bag-of-Lies [32], another dataset we will test our method on. We split the video into 20 chunks and select a single representative frame from each chunk. We form a vector by extracting Local Binary Pattern (LBP) [37] features for each frame and concatenate them in the order they appear in the video as proposed in [32]. This combined feature is then used further for classification using Support Vector Machine (SVM) [38], Random Forest [39] and Multilauer Perceptron (MLP) [40]. We compare with and report the results of the best model out of those three.

As a second baseline, we use Temporal Segment Network (TSN) [9], which is a two-stream neural network that utilizes a sparse temporal sampling strategy and video-level supervision to enable learning using the whole video. For each segment sampled from the video, TSN inputs an RGB image to the spatial stream and 5 Optical Flow maps to the temporal stream. Then, the outputs of each segment are combined using a consensus function H (such as softmax) to get the final video-level prediction. To compare TSN with our method we utilize their publicly available code [4].

Comparative Results We compare our method with the state-of-the-art alternative [32] as well as with prior approaches [10–12,35,36]. Most of these methods are multi-modal. Thus, to compare with them on equal terms, we report their results by using only visual cues. Those comparative results are given in Table 1.

From Table 1, we can see that our method outperforms all the other methods on the Real-Life Trial Dataset [10]. This validates our hypothesis that the performance of a model trained for video-based deception detection would benefit by taking as input higher-level features, instead of raw videos. We showed that by using 1-dimensional features, we can create a model that is simple and easy to train, yet perform better than previous approaches.

Table 1. Comparative results (%) on the Real-Life Trial Dataset [10]. Note that for all methods we use results reported only with visual cues, even if multi-modal results are given as well. For methods indicated with an *, we report the results directly from their papers. For † the authors use meta learning and adversarial learning. We do not perform any data augmentation.

Method	ACC	AUC
LBP [37]	75.00	76.15
TSN [9]	77.55	81.78
[36]*	67.20	-
[10]*	68.59	-
[11]*	75.42	-
[35] *	78.58	-
[12]*	-	83.47
FFCSN [13] *	89.16	91.89
FFCSN [13] * †	*93.16*	*96.71*
Ours	**92.36**	**97.27**

We wish to note that in [13] the authors also implement a version of their model that includes a Meta Learning and an Adversarial Learning [14,15] module

[4] https://github.com/yjxiong/tsn-pytorch

to compact the data scarcity of the dataset. We have different research targets and opt not to apply any method of data augmentation in our model. Thus, our models are not directly comparable.

Ablation Study Results We conduct an ablation study on the input features of our proposed method. In particular, we measure the performance of our model by using (i) only Gaze signals, (ii) only FAUs signals and (iii) FAUs + Gaze signals. The results are shown on Table 2. As expected, we notice that our method performs better using FAU signals than using just Gaze signals. However, we notice the best performance when those Gaze and FAU features are combined, meaning that FAU and Gaze signals are complementary.

Table 2. Ablation study results (%).

Modality	ACC	AUC
Gaze	79.73	83.05
FAUs	91.15	95.43
FAUs + Gaze	**92.36**	**97.27**

4.3 Experiments on Bag-of-Lies

We also use the *Bag-of-Lies* [32] dataset for evaluating our method. It consists of 35 subjects, each of whom is shown 6-10 images and then being asked to describe them. Each participant is free to describe the image honestly or deceptively and the answer is recorded in a video. The video recordings do not share the same length, they are ranging from 3.5 seconds to 42 seconds.

The total number of samples in the dataset is 325 with an even distribution of truth (163) and lie (162). Although this dataset offers information about other modalities as well (audio and EEG signals), we will use only the visual modality in our experiments.

Evaluation protocol. To evaluate our method, we use the same protocol as in [32]. We perform 3-fold cross validation across participants (12, 12 and 11 participants in each fold). We use the same metrics for the evaluation of deception detection methods as in the *Real-Life Trial* Dataset. Similarly, the results reported are average of cross validation over folds.

Baselines. For evaluating our method on the *Bag-of-Lies* dataset [32] we use the same baselines we used for the Real-Life Trial dataset. Our implementation using LBP features matched the results reported in [32].

Results Table 3 shows the results of our method and our baselines. We can see that our proposed method clearly outperforms all of them. Since the *Bag-of-Lies* dataset is recently introduced there are no other methods that report results in it. One thing to note is that this dataset is harder compared to the *Real-Life Trial* Dataset.

Table 3. Comparative results (%) on the Bag-of-Lies Dataset [32].

Method	ACC	AUC
LBP [37]	55.12	55.32
TSN [9]	56.94	57.62
Ours	**64.47**	**67.08**

4.4 Experiments on the Resistance Game

The dataset we use contains a set of videos that capture a group of 5-8 people, while playing a version of the *Resistance Game*, a social role-playing game.

Players in the Resistance Game are randomly given one of two roles, deceivers or truth-tellers. Deceivers know who have the same role as them in the game, whereas truth-tellers are clueless. The majority of players in a game are truth-tellers; there are 2-3 deceivers. The game proceeds in rounds or missions as they are called in the game. There are 3 to 7 missions in a game.

The players should nominate and elect a leader, who in turn nominates team members to go on a mission. All the players vote if that particular team should go on a mission or if a different team should be chosen. When on an mission, the team members vote in secret for the success or failure of the mission. The deceivers want the mission to fail, while the truth-tellers want it to succeed. When a mission succeeds all truth-tellers get a point, whereas when it fails the deceivers get one point. The team with the highest score at the end of the game wins. The game relies on deceivers trying to hide their identity and attempt to prevent the larger group from working together to reveal the deceivers among the group. Furthermore, it gives players a lot of opportunities to exhibit deceptive behaviors.

The dataset contains a set of videos involving 285 players collected from 5 sites in 3 different countries to account for possible heterogeneity in deceptive behavior among different cultures. The videos used are very long and their average duration is 46 minutes. In our experiments we used a balanced subset of the dataset, containing 230 videos.

Evaluation protocol. There are no 2 data points in our dataset that contain the same user. Thus, we perform 5-fold cross validation across videos. To evaluate our approach, we use the same metrics and baselines with the previous datasets.

Results In Table 4, we can see the results of our method and our baselines. *The Resistance* dataset is very difficult, which can be noticed by the fact that

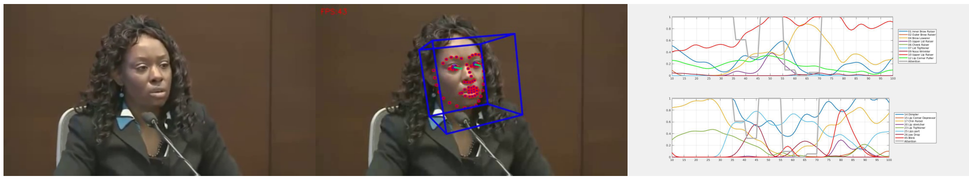

Fig. 3. The Real-Life Trial [10] dataset: (**left**) Screenshot of frames from original videos; (**middle**) facial landmark and head-pose bounding box visualizations by OpenFace [21]; (**right**) FAU waveforms and attention visualizations of the predictor model.

both baselines perform no better than a random classifier in that dataset. We speculate that this happens since we do not have enough data. The input videos are very long and the supervisory signal is weak (only 1 label for the whole video). As a result, both baselines overfit to background noise and fail miserably.

Nonetheless, our approach achieves substantially better performance than both baselines. This can be attributed to the fact that our method uses FAU and Gaze signals as input, which can be beneficial for the learning procedure. The model is asked to learn parameters that model the correlations of high-level facial information, and is robust to background changes and other noisy information present in the videos.

Table 4. Results (%) of our method on the Resistance Game Dataset.

Method	ACC	AUC
LBP [37]	49.56	49.56
TSN [9]	51.15	51.15
Ours	**71.08**	**71.08**

5 Attention Visualization

We use the Attention Module proposed in Section 3 to detect the *key frames* of the input video. Using this method, we can analyze the frames that our models considers important for classifying a video as one that contains deceptive behavior. We can treat the attention as an extra 1-D signal and visualize it along with the other waveforms to search for consistent patterns in deceptive behaviors. An example of such visualization is shown in Figure 3.

From our experiments, we can conclude that the facial signals constitute useful cues for detecting deception in videos. We think that our Attention Module can be used as a tool to quantitatively study the role of those facial signals as deception indicators.

6 Conclusion

We propose a novel framework for video-based deception detection and analysis of deceptive behavior. By using 1-dimensional FAU and Gaze signals, we are able to train a conceptually simple, modular and powerful model that performs really well in practice. Comprehensive evaluation results illustrate that our model achieves state-of-the-art performance, even though it far less intricate than previous approaches. This highlights the usefulness of facial information for the task of non-verbal deception detection. Finally, we propose a novel approach to interpret our model predictions, by computing the attention of the video classification model used. The Attention Module can be proven a useful tool for retrospective analysis of deceptive behavior by domain experts.

References

1. L. Tran and X. Liu, "Nonlinear 3d face morphable model," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7346–7355, 2018.
2. D. B. Buller and J. K. Burgoon, "Interpersonal Deception Theory," *Communication Theory*, vol. 6, no. 3, pp. 203–242, 03 1996. [Online]. Available: https://doi.org/10.1111/j.1468-2885.1996.tb00127.x
3. J. Burgoon, L. Guerrero, and K. Floyd, *Nonverbal communication*, 1st ed. Allyn Bacon, 2010.
4. P. Ekman and E. L. Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, 1997.
5. J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, April 2017.
6. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4489–4497.
7. S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan 2013.
8. K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 1, 06 2014.
9. L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2740–2755, Nov 2019.
10. V. Pérez-Rosas, M. Abouelenien, R. Mihalcea, and M. Burzo, "Deception detection using real-life trial data," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15. New York, NY, USA: ACM, 2015, pp. 59–66. [Online]. Available: http://doi.acm.org/10.1145/2818346.2820758
11. V. Pérez-Rosas, M. Abouelenien, R. Mihalcea, Y. Xiao, C. Linton, and M. Burzo, "Verbal and nonverbal clues for real-life deception detection," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2336–2346. [Online]. Available: https://www.aclweb.org/anthology/D15-1281

12. Z. Wu, B. Singh, L. Davis, and V. S. Subrahmanian, “Deception detection in videos,” *AAAI*, pp. 1695–1702, 2018.
13. M. Ding, A. Zhao, Z. Lu, T. Xiang, and J.-R. Wen, “Face-focused cross-stream network for deception detection in videos,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
14. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
15. A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *CoRR*, vol. abs/1511.06434, 2015.
16. J. Gehring, M. Auli, D. Grangier, and Y. Dauphin, “A convolutional encoder model for neural machine translation,” 01 2017, pp. 123–135.
17. N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, “Neural machine translation in linear time,” 2016. [Online]. Available: https://arxiv.org/abs/1610.10099
18. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499
19. S. Bai, J. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 03 2018.
20. Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019. [Online]. Available: https://doi.org/10.1109/TASLP.2019.2915167
21. T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. Morency, “Openface 2.0: Facial behavior analysis toolkit,” in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, May 2018, pp. 59–66.
22. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML’15. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167
23. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.
24. F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *CoRR*, vol. abs/1511.07122, 2015.
25. C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” *arXiv preprint arXiv:1505.05424*, 2015.
26. A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in *Advances in neural information processing systems*, 2017, pp. 5574–5584.
27. D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

28. Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, 2016, pp. 1050–1059.
29. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
30. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
31. L. Wang, Z. Wu, S. Karanam, K.-C. Peng, R. V. Singh, B. Liu, and D. N. Metaxas, “Sharpen focus: Learning with attention separability and consistency,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 512–521.
32. V. Gupta, M. Agarwal, M. Arora, T. Chakraborty, R. Singh, and M. Vatsa, “Bag-of-lies: A multimodal dataset for deception detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
33. K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *IEEE International Conference on Computer Vision (ICCV 2015)*, vol. 1502, 02 2015.
34. A. Baydin, R. Cornish, D. Rubio, M. Schmidt, and F. Wood, “Online learning rate adaptation with hypergradient descent,” 03 2017.
35. M. Gogate, A. Adeel, and A. Hussain, “Deep learning driven multimodal fusion for automated deception detection,” in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017, pp. 1–6.
36. M. Jaiswal, S. Tabibu, and R. Bajpai, “The truth and nothing but the truth: Multimodal analysis for deception detection,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 938–943.
37. T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.
38. C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, p. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411
39. Tin Kam Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, Aug 1995, pp. 278–282 vol.1.
40. G. E. Hinton, *Connectionist Learning Procedures.* IEEE Press, 1990, p. 11–47.